

Claims:

1. A system for the concurrent operation of plural applications, said system comprising:
 - (a) a shared object space capable of storing at least one shared object accessible to at least two said applications so that at least one said application is capable of causing an event in said object, said at least one application running in a virtual machine; and
 - (b) a listener operably associated with at least one said application running in a virtual machine, said listener listening for said event.
2. The system of claim 1 where said listener is located in said shared object space.
3. The system of claim 1 where at least two applications are capable of causing said event.
4. The system of claim 1 including plural said listeners.
5. The system of claim 1 where at least one of said virtual machines is a Java virtual machine.
6. The system of claim 1 where said shared object space is linked to each said plural application by a native method interface.
7. The system of claim 6 where said system includes a default directory with a native library.
8. The system of claim 1 including a non-object oriented application.

9. The system of claim 8 where said non-object-oriented application is a C application.
10. The system of claim 1 where access to said at least one object by said plural applications is synchronized.
11. The system of claim 1 where said plural applications implement a stock trading system.
12. The system of claim 1 where said at least one object is copy shared among said plural applications.
13. The system of claim 1 where said at least one object is direct shared among said plural applications.
14. The system of claim 1 including a system manager that analyzes information pertaining to the operation of said shared object space.
15. The system of claim 1 including a global name space in said shared object space.
16. A system for the concurrent operation of plural applications, said system comprising:
 - (a) a shared object space capable of storing at least one shared object accessible to at least two said applications so that at least one said application is capable of causing an event in said object, said at least one application running in a virtual machine;
 - (b) a listener operably associated with at least one said application running in a virtual machine, said listener listening for said event; and
 - (c) said listener being able to identify which application caused said event.

17. The system of claim 16 where said listener is located in said shared object space.

18. The system of claim 16 where at least two applications are capable of causing said event.

19. The system of claim 16 including plural said listeners.

20. The system of claim 16 where at least one of said virtual machines is a Java virtual machine.

21. The system of claim 16 where said shared object space is linked to each said plural application by a native method interface.

22. The system of claim 21 where said system includes a default directory with a native library.

23. The system of claim 16 including a non-object-oriented application.

24. The system of claim 23 where said non-object-oriented application is a C application.

25. The system of claim 16 where access to said at least one object by said plural applications is synchronized.

26. The system of claim 16 where said plural applications implement a stock trading system.

27. The system of claim 16 where said at least one object is copy shared among said plural applications.

28. The system of claim 16 where said at least one object is direct shared among said plural applications.

29. The system of claim 16 including a system manager that analyzes information pertaining to the operation of said shared object space.

30. The system of claim 16 including a global name space in said shared object space.

31. A system for the concurrent operation of plural applications, said system comprising:

(a) a shared object space capable of storing at least one shared object accessible to at least two said applications so that at least one said application is capable of causing an event in said object, said application running in a virtual machine and said object containing a header capable of containing an identifier of the application that caused said event;

(b) a listener operably associated with at least one said application running in a virtual machine, said listener listening for said event; and

(c) said listener being able to identify which application caused said event from said header.

32. The system of claim 31 where said listener is located in said shared object space.

33. The system of claim 31 where at least two applications are capable of causing said event.

34. The system of claim 31 including plural said listeners.

35. The system of claim 31 where at least one of said virtual machines is a Java virtual machine.

36. The system of claim 31 where said shared object space is linked to each said plural application by a native method interface.

37. The system of claim 36 where said system includes a default directory with a native library.

38. The system of claim 31 including a non-object-oriented application.

39. The system of claim 38 where said non-object-oriented application is a C application.

40. The system of claim 31 where access to said at least one object by said plural applications is synchronized.

41. The system of claim 31 where said plural applications implement a stock trading system.

42. The system of claim 31 where said at least one object is copy shared among said plural applications.

43. The system of claim 31 where said at least one object is direct shared among said plural applications.

44. The system of claim 31 including a system manager that analyzes information pertaining to the operation of said shared object space.

45. The system of claim 31 including a global name space in said shared object space.